IN THE CLAIMS

Claims 1-6 (Cancelled).

7. (Previously Presented) A method comprising:

(a) identifying a conditional branch instruction in a program;

(b) associating a condition for the identified conditional branch instruction with one or more instructions in one or more branch instruction paths for the conditional branch instruction to produce one or more conditional instructions; and

(c) executing each conditional instruction, the executing comprising:

(i) performing both a first operation from a first branch instruction path to produce a first result and a second operation from a second branch instruction path to produce a second result,

(ii) associating both the first result and the second result with the condition,

(iii) outputting both the first result and the second result with the associated condition, and

(iv) retiring the first result or the second result based on how the condition is resolved.

8. (Original) The method of claim 7, wherein the performing comprises performing the first operation using at least one operand for the conditional instruction to produce the first result and performing the second operation to produce an operand for the conditional instruction as the second result.

9. (Original) The method of claim 7, wherein the identifying comprises identifying a conditional forward branch instruction.

10. (Original) The method of claim 7, wherein the identifying comprises identifying a conditional branch instruction having a fall through branch instruction path having less than or equal to a predetermined number of instructions and/or a conditional branch instruction having a target branch instruction path having less than or equal to a predetermined number of instructions.

11. (Original) The method of claim 7, wherein the identifying comprises identifying a conditional branch instruction that is not predictable within a predetermined degree of accuracy.

12. (Original) The method of claim 7, wherein each conditional instruction is a conditional micro-operation and wherein the method comprises decoding one or more macro-instructions in the branch instruction path to produce one or more conditional micro-operations.

13. (Previously Presented) The method of claim 7, comprising committing the first result or the second result based on how the condition is resolved.

14. (Previously Presented) The method of claim 7, comprising:

(d) executing another instruction dependent on the executed instruction, the executing the other instruction comprising:

(i) performing an operation designated by the other instruction on the first result to produce a third result and on the second result to produce a fourth result,

(ii) associating both the third result and the fourth result with the condition, and

(iii) outputting both the third result and the fourth result with the associated condition.

15. (Original) The method of claim 14, comprising retiring the third result or the fourth result based on how the condition is resolved.

16. (Previously Presented) A processor comprising:

an operation unit to perform both a first operation from a first instruction path designated by a conditional instruction to produce a first result, and a second operation from a second instruction path designated by the conditional instruction to produce a second result;

a memory to store a condition associated with the conditional instruction in association with the first result and the second result; and

a retirement unit to retire either the first result or the second result based on how the condition is resolved.

17. (Original) The processor of claim 16, comprising a conditional branch processing unit to associate a condition for a conditional branch instruction with one or more instructions in one or more branch instruction paths for the conditional branch instruction to produce one or more conditional instructions.

18. (Original) The processor of claim 16, wherein the conditional instruction is a conditional micro-operation, and wherein the processor comprises a decoder to decode macro-instructions of a program into micro-operations.

Claims 19-21 (Cancelled).

22. (Currently Amended) A processor comprising:

(a) a dispatch/execute unit to dispatch and execute instructions for a program out of order, the dispatch/execute unit comprising an execution unit to perform both a first operation from a first instruction path designated by a conditional instruction to produce a first result and a second operation from a second instruction path designated by the conditional instruction to produce a second result and to associate both the first result and the second result with a condition associated with the conditional instruction;

(b) a reorder buffer to store results of executed instructions; and

(c) a retire unit to retire results of executed instructions, the retire unit to retire the first result or the second result based on how the condition is resolved.

23. (Original) The processor of claim 22, comprising a conditional branch processing unit to associate a condition for a conditional branch instruction with one or more instructions in one or more branch instruction paths for the conditional branch instruction to produce one or more conditional instructions.

24. (Original) The processor of claim 22, comprising a fetch/decode unit to fetch macro-instructions of the program and to decode the fetched macro-instructions into micro-operations, wherein the dispatch/execute unit dispatches and executes micro-operations.

25. (Original) The processor of claim 22, the execution unit to perform an operation designated by another instruction on a first result of a conditional instruction to produce a third result and on a second result of a conditional instruction to produce a fourth result and to associate both the third result and the fourth result with a condition.

26. (Currently Amended) A computer system comprising:

(a) a memory to store instructions of a program;

(b) a processor to perform both a first operation from a first instruction path designated by a conditional instruction to produce a first result and a second operation from a second instruction path designated by the conditional instruction to produce a second result, to associate both the first result and the second result with a condition associated with the conditional instruction, to output both the first result and the second result with the associated condition and to retire the first result or the second result based on how the condition is resolved.

27. (Original) The computer system of claim 26, the processor to perform an operation designated by another instruction on a first result of a conditional instruction to produce a third result and on a second result of a conditional instruction to produce a fourth result and to associate both the third result and the fourth result with a condition.

28. (Original) The computer system of claim 26, wherein the processor comprises:

(i) a fetch/decode unit to fetch macro-instructions of the program and to decode the fetched macro-instructions into micro-operations;

(ii) a dispatch/execute unit to dispatch and execute the micro-operations out of order, the dispatch/execute unit comprising an execution unit to perform the first operation and the second operation and to output the first result and the second result in association with the condition;

(iii) a reorder buffer to store results of executed micro-operations; and

(iv) a retire unit to retire results of executed micro-operations, the retire unit to retire the first result or the second result based on how the condition is resolved.

29. (Original) The computer system of claim 26, wherein the processor comprises a conditional branch processing unit to associate a condition for a conditional branch instruction with one or more instructions in one or more branch instruction paths for the conditional branch instruction to produce one or more conditional instructions.

30. (Previously Presented) The computer system of claim 26, comprising:
a memory controller hub to provide an interface to the memory; and
an input/output controller hub coupled to the memory controller hub.